

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/SG04/000411

International filing date: 15 December 2004 (15.12.2004)

Document type: Certified copy of priority document

Document details: Country/Office: IN
Number: 338/CHE/2004
Filing date: 15 April 2004 (15.04.2004)

Date of receipt at the International Bureau: 12 April 2005 (12.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse



GOVERNMENT OF INDIA
PATENT OFFICE
Ministry of Commerce and Industry
Department of Industrial Policy and Promotion

It is hereby certified that annexed here to is a true copy of **Application, Provisional Specification & Drawings** of the patent application as filed and detailed below:-

Date of application : 15-04-2004

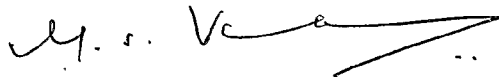
Application No : 338/CHE/2004

Applicants : M/s. Matrix View Technologies (India) Private Limited,
No. 69, Mahalakshmi Koil Street, Kalakshetra Colony,
Besant Nagar, Chennai – 600 090. India
an Indian Company

In witness there of
I have here unto set my hand

Dated this the 18th day of March 2005
27th day of Phalguna, 1926(Saka)

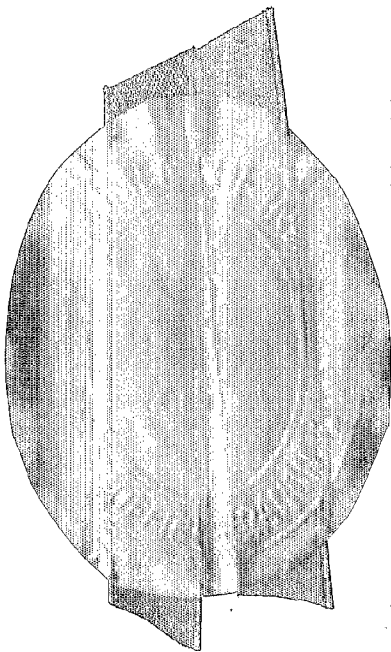
By Authority of
**THE CONTROLLER GENERAL OF PATENTS,
DESIGNS AND TRADE MARKS.**



(M.S.VENKATARAMAN)
ASSISTANT CONTROLLER OF PATENTS & DESIGNS



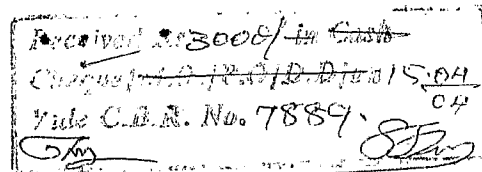
PATENT OFFICE BRANCH
Guna Complex, 6th Floor, Annex.II
No.443, Anna Salai, Teynampet,
Chennai – 600 018. India.



FORM 1

THE PATENTS ACT, 1970
(39 OF 1970)

APPLICATION FOR GRANT OF A PATENT
(SEE SECTIONS 5(2), 7, 54 AND 135 AND RULE 39)



15/4

ORIGINAL
338/CHE/2004
15 APR 2004

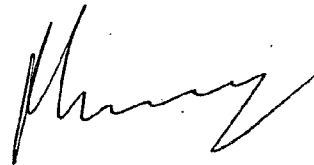
1. WE, MATRIXVIEW TECHNOLOGIES (INDIA) PRIVATE LIMITED,
of NO. 69, MAHALAKSHMI KOIL STREET,
KALAKSHETRA COLONY, BESANT NAGAR, CHENNAI -600090,
INDIA
AN INDIAN COMPANY
2. hereby declare -
 - (a) that we are in possession of an invention titled
"OPTIMISING COMPRESSION FOR IMAGE DATA"
 - (b) that the Provisional Specification relating to this invention is filed with this application.
 - (c) that there is no lawful ground of objection to the grant of a Patent to us.
3. We further declare that the inventors for the said invention is/are :

NAME (a)	ADDRESS (b)	NATIONALITY (c)
THIAGARAJAN ARVIND	H24/6, VAIGAI STREET, BESANT NAGAR, 600090 CHENNAI, TAMIL NADU, INDIA	INDIAN
4. That we are assignees of the inventors.
5. That our address for service in India is as follows:- D. P. AHUJA & CO., 53 Syed Amir Ali Avenue, Calcutta 700 019, West Bengal, India. TEL: (033)22819195, FAX: (033)24757524.
6. That to the best of our knowledge, information and belief the fact and matters stated herein are correct and that there is no lawful ground of objection to the grant of patent to us on this application.
7. Following are the attachments with the application:
 - (a) Provisional Specification (2 copies)
 - (b) Statement and Undertaking on Form 3 in duplicate
 - (c) Formal drawings (14 sheets) (Provisional) in duplicate
 - (d) Rs 3,000/- by cheque bearing No.906540 dated 13.04.2004 on ICICI BANK.

Contd...2

We request that a patent may be granted to us for the said invention.

Dated this 13th day of April, 2004.



(S.D. AHUJA)
OF D. P. AHUJA & CO
APPLICANTS' AGENT

To
The Controller of Patents,
The Patent Office,
Chennai

FORM 2

**THE PATENTS ACT, 1970
(39 of 1970)**

**PROVISIONAL SPECIFICATION
(See Section 10)**

TITLE

OPTIMISING COMPRESSION FOR IMAGE DATA

APPLICANT

**MATRIXVIEW TECHNOLOGIES (INDIA) PRIVATE LIMITED,
of NO. 69, MAHALAKSHMI KOIL STREET, KALAKSHETRA COLONY,
BESANT NAGAR, CHENNAI -600090,
INDIA
AN INDIAN COMPANY**

The following specification particularly describes the nature of the invention and the manner in which it is to be performed.

15 APR 2004 338 /CHE/2004 ORIGINAL

Field of the Invention

5

The present invention relates to a method and system for optimising the compression of image data and other highly correlated data streams.

Background of Invention

10

Image and data compression is of vital importance and has great significance in many practical applications. To choose between lossy compression and lossless compression depends primarily on the application.

15

Some applications require a perfectly lossless compression scheme so as to achieve zero errors in the automated analysis. This is particularly relevant when where an automatic analysis is performed on the image or data. Generally, Huffman coding and other source coding techniques are used to achieve lossless compression of image data.

20

In certain other applications, the human eye visually analyzes images. Since the human eye is insensitive to certain patterns in the images, such patterns are discarded from the original images so as to yield good compression of data. These schemes are termed as "visually lossless" compression schemes. This is not a perfectly reversible process as the de-compressed image data is different from the original image data. The degree of difference depends on the quality of compression, and the compression ratio. Compression schemes based on discrete cosine transforms (DCT) and Wavelet transforms followed by lossy quantization of data are typical examples of visually lossless scheme.

30

As a general rule, it is desirable to achieve the maximum compression ratio with zero, or minimal, possible loss in the quality of the image. At the same time, the complexity involved in the system and the power consumed by the image compression system are important parameters when it comes to a hardware-based implementation.

35

Usually, image compression is carried out in two steps. The first step is to use a pre-coding technique, which is normally based on signal transformations. The second step would be to further compress the data values by standard source coding techniques such as, for example, Huffman and Lempel-Ziv schemes.

- Most efficient compression techniques require a transformation. This is also known as pre-coding. The initial pre-coding step is the most critical and important operation in image compression. The complexity involved with DCT and Wavelet based transformations is quite high because of the large number of multiplications involved. This is illustrated in the following DCT equation:

$$DCT(i, j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right]$$

$$\text{where } C(x) = \frac{1}{\sqrt{2}} \text{ if } x=0, \text{ else } 1 \text{ if } x>0.$$

- In addition to the large number of multiplications involved in carrying out the above DCT equation, there is also a zigzag rearrangement of the image data, which involves additional complexity. DCT transformation uses a mathematical algorithm to generate frequency representations of a block of video pixels. DCT is an invertible, discrete orthogonal transformation between time and frequency domain.

- Transformation aids in increasing the efficiency of a second step, the entropy coder. At this stage, if the entropy coder produces good compression ratios, then the pre-coding should transform the data into a form suitable for the entropy coder. If the transformation is not efficient, then the entropy coder becomes redundant. Thus, pre-coding is the most important stage of any image compression algorithm.

- Another important property of any transformation is that it is reversible, to allow the reverse process to be applied at the decompression stage to obtain the original image. This transformation is extensively used in JPEG algorithms and their variants.

- However, DCT suffers from several problems. Firstly, the complexity of the equation in terms of the number of multiplications and additions. In the 2D case, with an array of dimension $N \times N$, the number of multiplications is in the order of $2N^3$ using a separable approach of computing 1D row and column DCTs. Specifically, for an 8×8 pixel array which is used in the JPEG family, 1024 multiplications and 896 additions are required. There have not been any significant improvements to reduce this computational overhead.

Even though the image data is an integer, their multiplication to cosine terms in the formula produces fractional numbers or real numbers because cosine values are

fractional in nature until and unless they are integer multiples of π , which may not be the case. Since fractional numbers need infinite precision to store them exactly, they might produce errors in the reverse process, resulting in loss.

- 5 Another popular transformation is the wavelet transform. This is used, for example, in the JPEG2000 image compression standard. A mother wavelet is used to decompose the image data into frequency sub-bands, which in turn increases the redundancy in most of the sub-bands, thereby improving compression ratios. Used in their original form, the mother wavelets do not give integer-to-integer transformation but when used
10 after a process called lifting, they come integer-to-integer transforms. This makes the entire process lossless but does not achieve a high compression ratio.

- Color transformations also offer another compression technique. A commonly used color space is RGB. In RGB, every pixel is quantized by using a combination of Red,
15 Green and Blue values. This format is popular among graphic designers, but is not ideal as a compression algorithm.

- It is desirable to provide an image compression system which does not involve rigorous transforms, and complex calculations. It also has to be memory efficient and
20 power efficient.

There are various image compression techniques presently available. A familiar few are JPEG, JPEG-LS, JPEG-2000, CALIC, FRACTAL and RLE.

- 25 JPEG compression is a trade-off between degree of compression, resultant image quality, and time required for compression/decompression. Blockiness results at high image compression ratios. It produces poor image quality when compressing text or images containing sharp edges or lines. Gibb's effect is the name given to this phenomenon - where disturbances/ripples may be seen at the margins of objects with
30 sharp borders. It is not suitable for 2-bit black and white images. It is not resolution independent, and does not provide for scalability, where the image is displayed optimally depending on the resolution of the viewing device.

- JPEG-LS does not provide support for scalability, error resilience or any such
35 functionality. Blockiness still exist at higher compression ratios and it does not offer any particular support for error resilience, besides restart markers.

JPEG-2000 does not provide any truly substantial improvement in compression efficiency and is significantly more complex than JPEG, with the exception of JPEG-LS

for lossless compression. The complexity involved in JPEG-2000 is higher for a lower enhancement in the compression ration and efficiency.

5 Although CALIC provides the best performance in lossless compression, it cannot be used for progressive image transmission as it implements a predictive-based algorithm that can work only in lossless/nearly-lossless mode. Complexity and computational cost are high.

10 These conventional schemes for image compression are not very well suited for hardware-based implementation.

Summary of the Invention

15 In a first preferred aspect, there is provided a method for optimising compression of image data of an image, comprising:

transforming the image data; and

rearranging the transformed image data by causing elements of the image data to be repetitive;

20 wherein the number of elements repeated is dependant upon a predetermined level of image quality selected for the compressed image.

The image data may be transformed according to Discrete Cosine Transform (DCT), wavelet transform or color transform.

25 The transformed image data may be rearranged by performing an end-to-end reversible sort together with a last to front transform.

30 The image data may originate from a photo, drawing or video frame. A portion of the image data may be compressed lossless while the remaining portion of the image data is compressed lossy.

The optimised image data for compression may be passed to an input of a source coder. The source coder may comprise an arithmetic coder preceded by a run-length encoder.

35

The method may further comprise additional compression of the image data wherein each element is compared with a previous element. If they are both equal, a first value is recorded. If they are not both equal, a second value is recorded. Each element may be a pixel. The first value may be a 1, and the second value may be a 0.

The first and second values may be stored in a bit plane. For a one-dimensional compression, a single bit plane may be used to store the values. However, for a two-dimensional compression, comparison may be in both horizontal and vertical directions, a separate bit plane being used for each direction.

The bit-planes for the horizontal and vertical directions may be combined by binary addition to form a repetition coded compression bit-plane. Combining may be by binary addition, only the second values being stored for lossless reconstruction of the image.

10 The result of the combining may be repetition coded compression data values. All other image data values may be able to be reconstructed using the repetition coded compression data values, and the bit planes for the horizontal and vertical directions.

Storage in bit planes may be in a matrix. A single mathematical operation may be performed for each element.

In another aspect, there is provided a system for optimising compression of image data of an image, comprising:

- a data transforming module to transform the image data; and
- 20 a data rearranging module to rearrange the transformed image data by causing elements of the image data to be repetitive,

wherein the number of elements repeated is dependant upon a predetermined level of image quality selected for the compressed image.

25 The system may further comprise: a camera for capturing at least one image and for supplying digital data; a reshaping block for rearranging the digital data into a matrix of image data values; a processor for receiving the matrix of image data values and compressing the image data values to form compressed data; and a memory for storage of the compressed data.

30

The camera may be analog. An analog-to-digital converter may be used to convert the analog image to provide the digital data.

35 The method may be used for an application selected from: medical image archiving, medical image transmission, database system, information technology, entertainment, communications applications, and wireless application, satellite imaging, remote sensing, and military applications.

Brief Description of the Drawings

In order that the invention may be fully understood and readily put into practical effect, there shall now be described by way of non-limitative example only a preferred embodiment of the present invention, the description being with reference to the accompanying illustrative drawings in which:

- Figure 1 is a process flow diagram of the optimisation process for compressing image data;
- 10 Figure 2 is a block diagram of a system for optimising compression of image data;
- Figure 3 illustrates a compression system based on repetition coded compression on a hardware implementation;
- Figure 4 is a sample grayscale image of a human brain, which is captured by magnetic resonance imaging ("MRI") to demonstrate the compression able to be achieved by repetition coded compression system;
- 15 Figure 5 is an enlarged image of a small region from Figure 4;
- Figure 6 shows that the image of Figure 2 is made up of many pixels in grayscale;
- Figure 7 shows a 36-pixel region within the sample MRI image of Figure 4;
- Figure 8 shows the ASCII value equivalent of the image data values for the image of Figure 4;
- 20 Figure 9 shows the application of repetition coded compression along the horizontal direction in the image matrix;
- Figure 10 shows the application of repetition coded compression along the vertical direction in the image matrix;
- 25 Figure 11 shows the combination of horizontal and vertical bit-planes by a binary addition operation;
- Figure 12 shows the total memory required for the 36-pixel region before and after applying repetition coded compression;
- Figure 13 shows the application of repetition coded compression to the entire image;
- 30 and
- Figure 14 shows the operational flow for the implementation of repetition coded compression.

Detailed Description of Preferred Embodiments

35

Image data is highly correlated. This means that more often than not adjacent data values in an image are repetitive in nature. Therefore, it is possible to achieve some compression out of this repetitive property of the image and then apply Huffman

coding or other source coding schemes. High compression ratios can be achieved by combining existing data transforms and source encoders.

5 The human eye is more sensitive to luminance than color. Thus, chrominance
luminance and value format offers another compression technique. This technique
uses color transformations in image compression to generate visually lossless
algorithms. Using lossy color transformation provides an effect equivalent to that of
quantization of other techniques in the sense that it cannot resolve the difference
10 between small values. That is, the same integer value is used for two different integer
values with a small difference. As a result of this, repetition occurs at a 24-bit level.
Increasing repetition in image data provides a high compression ratio. However, one
drawback to this technique is that it is not reversible perfectly, that is, it is lossy. In
other words, the decompressed image data is different from the original image data.
The degree of difference is dependent upon the quality of compression and also the
15 compression ratio. The adjustment of the quality may be user-defined by setting a
quality parameter such that a very highly compressed visually lossless image is
produced. By visually lossless we mean that the image data is technically lossy but to
the human eye the image appears lossless.

20 Referring to Figure 1, a method 50 for optimising compression of image data is
provided. The quality of the resultant compressed image is initially defined 51. This will
determine the amount of repetition to be artificially generated in the image data. A
higher amount of repetition means that a larger difference between adjacent pixels is
tolerated (more lossy). If these pixels differ below a certain level they are considered to
25 be identical. A lower amount of repetition means that the image is less lossy and
visually lossless. The pre-coding block of the process is divided into two logical stages
52, 53. The first stage is transformation 52. Transformation 52 can be any one of DCT,
wavelet or color transformations. The second stage is data rearrangement 53. After
the data is transformed and re-arranged, it is then directed 56 to the input of a source
30 coder. The source coder comprises an arithmetic coder preceded by a run length
encoder.

The data rearrangement stage 53 is primarily responsible for optimising the image
data for compression later. This optimisation consists of an end-to-end reversible sort
35 54 along with a last to front transform 55. The result is that the rearranged data
optimises compression by creating repetition to increase the compression ratio.

The optimisation process is scalable since the quality of the compressed image is user
defined 51 at run-time. The optimisation process does not require significant changes

to be made to the structure of the optimisation process. For example, when a large set of data is to be compressed into a limited amount of disk space, the choice of a compression ratio depends on the desired quality for individual images or a group of images. For Internet applications, such as streaming media and telephony applications, it is ideal for digital media developers to be able to define quality of the resultant compressed image by selecting the compression ratio.

Selected areas of an image rather than the entire image can be optimised for compression. For example, a selected region of the image can be compressed in a lossless manner, with the other regions of the image compressed in a lossy manner. This scenario is ideal for graphic artists that may want certain areas of their images to remain in perfect quality. The overhead complexity of optimising across the images is minimal, while significant gains in compression and quality are obtained.

High compression ratios are achieved while maintaining a reduced pixel-to-pixel error. The scalability of the optimisation process is maintained by exploiting the close correlation between adjacent pixels by artificially creating repetition.

Using the algorithm, a lower Mean Square Error (MSE) is achieved compared to JPEG, JPEG2000. In JPEG, the MSE is higher due to the quantization process. Also, the algorithm is a visually lossless algorithm where the pixel-to-pixel losses are smaller in order to deliver high compression ratios.

Referring to Figure 2, optimising the compression of image data is performed by an optimisation system 60. The system 60 comprises a data transforming module 61 to transform the image data and a data rearranging module 62 to rearrange the transformed image data by artificially generating repetition of elements of the image data. The level of repetition corresponds to a predetermined level of image quality for the compressed image. The rearranged data is passed to an input of a source coder 63. The source coder 63 comprises an arithmetic coder 65 preceded by a run length encoder 64.

Additional repetition coded compression (RCC) is applied 57 after the image data has been optimised for compression. In RCC, each element is compared with the previous element. If both of them are equal then a value of "1" is stored in a bit-plane. Otherwise a value of '0' is stored in the bit-plane. Only the difference value is stored in a matrix, instead of storing all the repeating values.

In a one-dimensional performance of the method, only one bit-plane is used to code the repetition in the horizontal direction.

5 But in a two-dimensional performance of the method, two bit-planes are used to code the repetitions in both the horizontal and the vertical directions. This is more efficient and gives a better compression ratio.

10 The RCC compression system is based on a mathematical comparison of adjacent image data values. The comparison is performed between adjacent image data values in both the horizontal as well as vertical directions. The bit-planes formed as a result of the comparison in the horizontal and vertical directions are respectively combined by a binary addition method. After this the resultant bit-plane positions are called as RCC bit-planes. The zero values in the RCC bit-plane are stored for lossless reconstruction of the original image. For lossless reconstruction, they are the only
15 values stored. The stored values correspond to the same locations in the original image matrix as zeros in the RCC bit-plane and are hereinafter called RCC data values. All the other image data values can be reconstructed by using the RCC data values, and the horizontal and vertical bit-planes.

20 In case of a lossy system of implementation, the adjacent pixels are not only compared for repetition, but also for the difference value. If the difference value between adjacent pixels is less than a given arbitrary threshold value, then the two adjacent pixels are made as the same. This further increases the number of repetitions in the image data and therefore also increases the compression ratio after repetition coded
25 compression is applied. The value of the threshold can be varied according to the requirements of the particular application, and system. The higher the threshold, the better the compression ratio and also the higher the loss in the quality of the reconstructed image.

30 Figure 3 illustrates the RCC compression system based on repetition coded compression on a hardware implementation. The analog image signals 12 are captured by the camera 10 and are converted into respective digital data 16 by an analog to digital converter 14. This digital data 16 is rearranged into a matrix of image data values by a reshaping block 18. The reshaped image matrix is stored in the
35 embedded chip 20, which performs the entire repetition coded compression system. This therefore gives the compressed repetition coded compression data values 22 and also the bit-planes of data 24 for storage, archival and future retrieval 26.

Figure 4 is a sample image of the human brain which is captured by magnetic resonance imaging (MRI). This sample image may be used to demonstrate the compression achieved by repetition coded compression. It is a grayscale image.

- 5 Figure 5 zooms a small region from the sample MRI image of the human brain. This zoomed region may also be used for demonstrating the repetition coded compression system.

Figure 6 shows that the image is made up of lot of pixels in grayscale.

10

Figure 7 shows a 36-pixel region within the sample MRI image of the human brain.

- Figure 8 shows the ASCII value equivalents of the image data values which are originally used for data storage. Each value requires eight bits (1 byte) of data memory. Currently, the 36-pixel region requires about 288 bits or 36 bytes of data memory. That data could be compressed and stored with only 112 bits after repetition coded compression.

15

- Figure 9 shows the application of repetition coded compression along the horizontal direction in the image matrix. This results in the horizontal bit-plane and also the horizontal values stored.

20

- Figure 10 shows the application of repetition coded compression along the vertical direction in the image matrix. This result in the vertical bit-plane, and also the vertical values stored.

25

Figure 11 shows the combination of horizontal and vertical bit-planes by a binary addition operation. This results in only five zero values which correspond to the final values stored from the original image matrix.

30

Figure 12 shows the total memory required for the 36-pixel region before and after applying repetition coded compression. The original memory requirement was 288 bits. After applying repetition coded compression the memory required was 112 bits. This is a great amount of compression.

35

Figure 13 shows the application of repetition coded compression to the entire image. The size is compressed to 44,000 bits from the original 188,000 bits.

Figure 14 shows an implementation of repetition coded compression. The image matrix 1201 is transposed 1202, encoded along the horizontal 1203 and vertical 1204 directions and the respective bit-planes 1205, 1206 are derived. Further compression is achieved by combining the horizontal and vertical bit-planes 1203, 1204 by a binary addition operation. This results in the repetition coded compression bit-plane 1207, which is logically inverted 1208 and compared 1209 with the original image matrix 1201 to obtain the final repetition coded compression data values 1210. The repetition coded compression data values 1210, together with the horizontal and vertical 1206 bit-planes are stored in a data memory 1211 for archival and future retrieval.

10

The coded data can be further compressed by Huffman coding. This compression of the image data is achieved using the repetition coded compression system. This system is fast as it does not make use of complex transform techniques. The method may be used for any type of image file. In the example given above, the system is applied only for grayscale images. It may be applied to color images.

15

The system of repetition coded compression of images may be applied to fields such as, for example, medical image archiving and transmission, database systems, information technology, entertainment, communications and wireless applications, satellite imaging, remote sensing, military applications.

20

RCC is based on a single mathematical operation and requires no multiplication for its implementation. This results in memory efficiency, power efficiency, and speed, in performing the compression. Because of the single mathematical operation involved, the system is reversible and lossless. This may be important for applications which demand zero loss. The compression ratios may be significantly higher than existing lossless compression schemes.

25

If the application permits a lossy compression system, a modification is made to the mathematical operation so that a certain amount of loss is observed in the compression, thereby resulting in higher compression ratios. This lossy compression system would find great applications in entertainment and telecommunication systems.

30

Applications

Repetition Coded Compression (RCC) can be used in applications for medical imaging, digital entertainment and document management. Each of these verticals requires RCC to be implemented in a unique way to deliver a robust and powerful end product.

35

RCC can be deployed in the following forms for commercialisation:

- 1) ASIC or FPGA chips
- 2) DSP or embedded systems
- 3) Standalone hardware boxes
- 5 4) Licensable software (as DLLs or OCX)
- 5) Software deliverables

Whilst there has been described in the foregoing description a preferred embodiment of the present invention, it will be understood by those skilled in the technology that
 10 many variations or modifications in details of design, constructions or operation may be made without departing from the present invention.

A method for optimising compression of image data of an image, comprising:

- transforming the image data; and
- 15 rearranging the transformed image data by causing elements of the image data to be repetitive;
- wherein the number of elements repeated is dependant upon a predetermined level of image quality selected for the compressed image.

20 The image data is transformed according to Discrete Cosine Transform (DCT), wavelet transform or color transform.

The transformed image data is rearranged by performing an end-to-end reversible sort together with a last to front transform.

25 The image data originates from a photo, drawing or video frame.

A portion of the image data is compressed lossless while the remaining portion of the image data is compressed lossy.

30 Rearranged data is passed to an input of a source coder.

The source coder comprises an arithmetic coder preceded by a run length encoder.

35 The method comprises additional compression of the rearranged image data wherein each element is compared with a previous element and:

- (a) if they are both equal, a first value is recorded; and
- (b) if they are not both equal, a second value is recorded.

Each element is a pixel.

The first value is a 1, and the second value is a 0.

5

The first and second values are stored in a bit plane.

For a one-dimensional compression, a single bit plane is used to store the values.

10

For a two-dimensional compression, comparison is in both horizontal and vertical directions, a separate bit plane being used for each direction.

The bit-planes for the horizontal and vertical directions are combined by binary addition to form a repetition coded compression bit-plane.

15

The combining is by binary addition, only the second values being stored for lossless reconstruction of the image.

20

The result of the combining is repetition coded compression data values, all other image data values being able to be reconstructed using the repetition coded compression data values, and the bit planes for the horizontal and vertical directions.

Storage in bit planes is in a matrix.

25

A single mathematical operation is performed for each element.

A system for optimising compression of image data of an image, comprising:

a data transforming module to transform the image data; and

30

a data rearranging module to rearrange the transformed image data by causing elements of the image data to be repetitive,

wherein the number of elements repeated is dependant upon a predetermined level of image quality selected for the compressed image.

35

The system comprises a source coder to receive the rearranged data as input.

the source coder comprises an arithmetic coder preceded by a run length encoder.

The system further comprising:

- (a) a camera for capturing at least one image and for supplying digital data;
- (b) a reshaping block for rearranging the digital data into a matrix of image data values;
- (c) a processor for receiving the matrix of image data values and compressing the image data values to form compressed data; and
- (d) a memory for storage of the compressed data.

The camera is analog, an analog-to-digital converter being used to convert the analog image to provide the digital data.

The method is used for an application selected from the group consisting of: medical image archiving, medical image transmission, database system, information technology, entertainment, communications applications, and wireless application, satellite imaging, remote sensing applications.

The predetermined level of image quality is user defined.

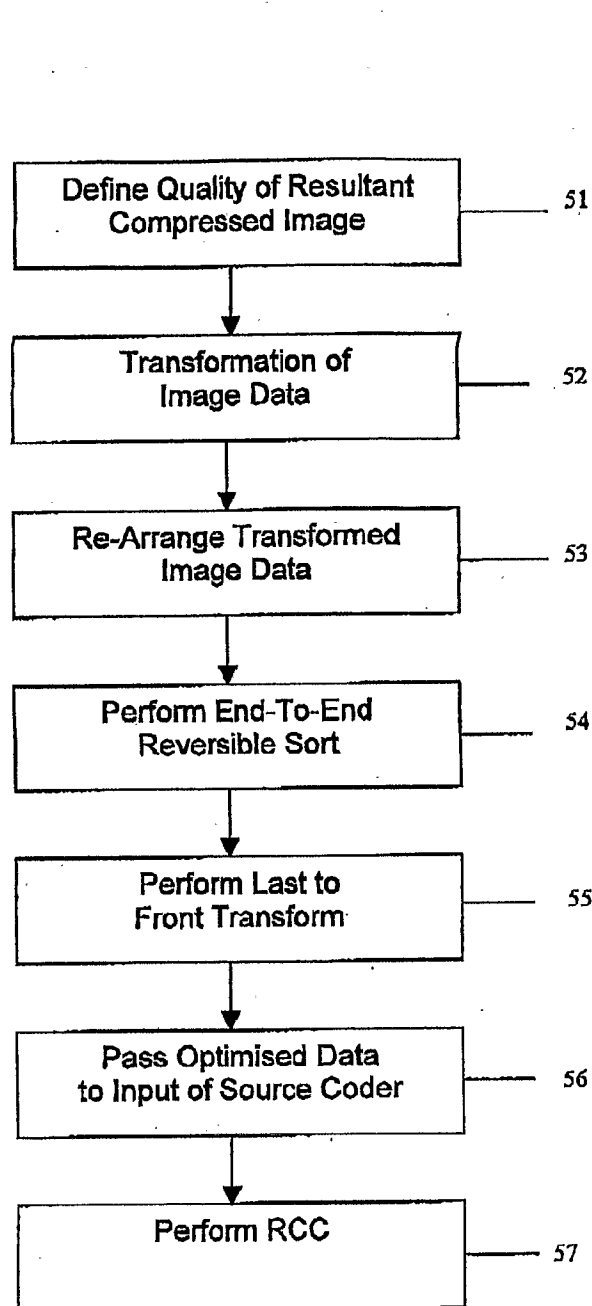
Dated This 13th Day of April 2004



(S.D.AHUJA)

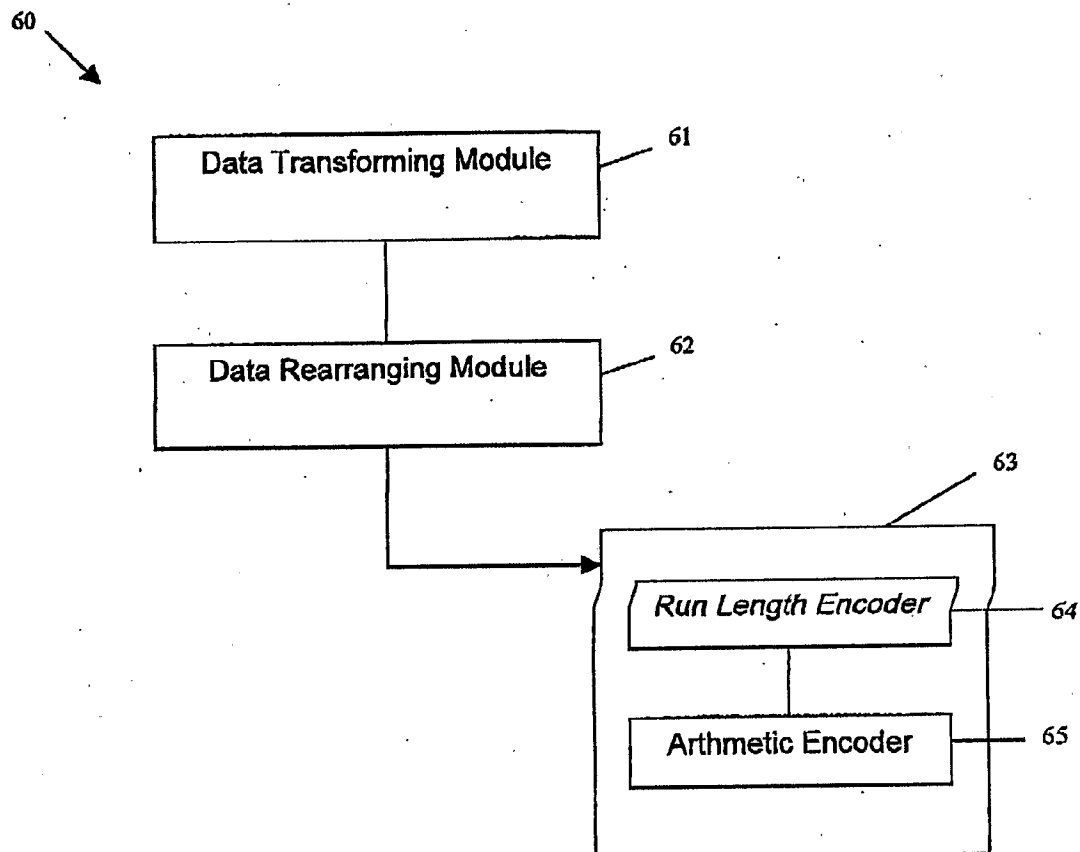
OF D.P.AHUJA&CO
APPLICANT'S AGENT

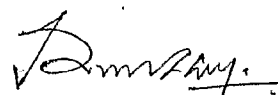
PROVISIONAL

**Figure 1**

Soumen Mukherjee
(SOUMEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

PROVISIONAL

**Figure 2**


(SOUMEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

RCC System for Hardware Implementation

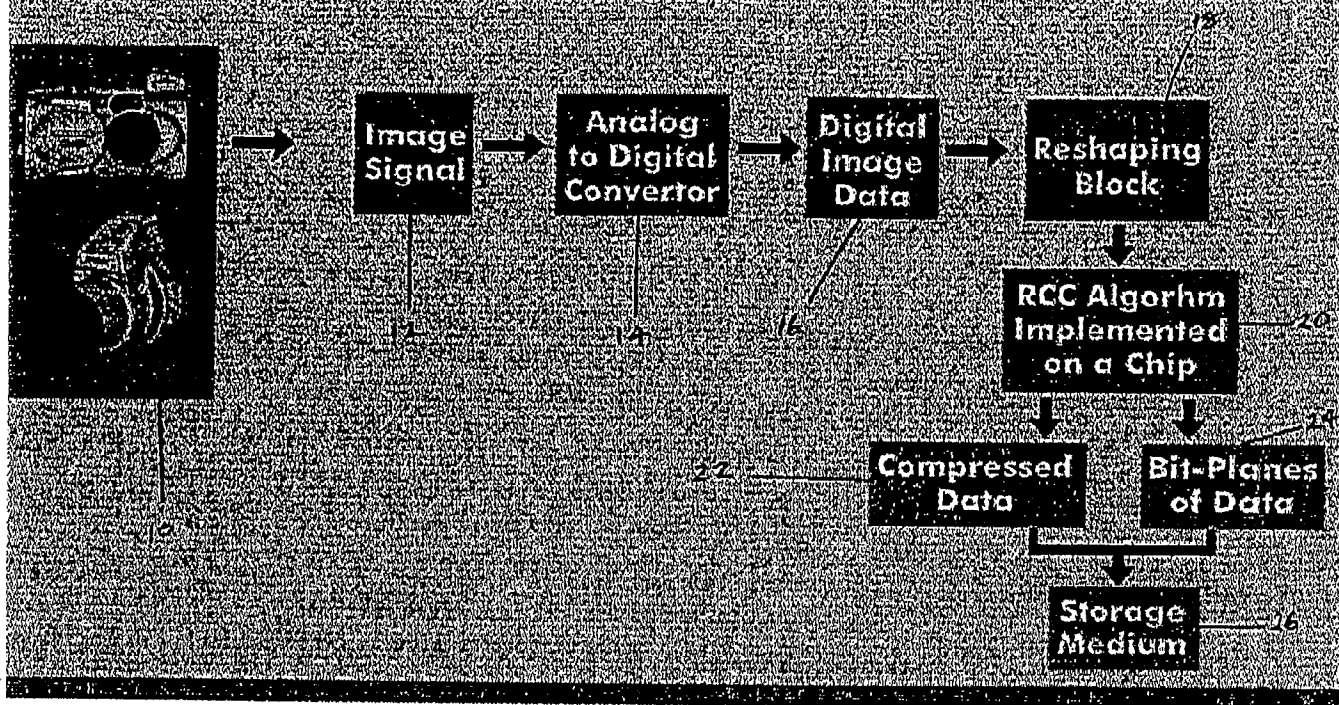


Figure 3

(Signature)
(SOUMEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

Sample MRI of Human Brain

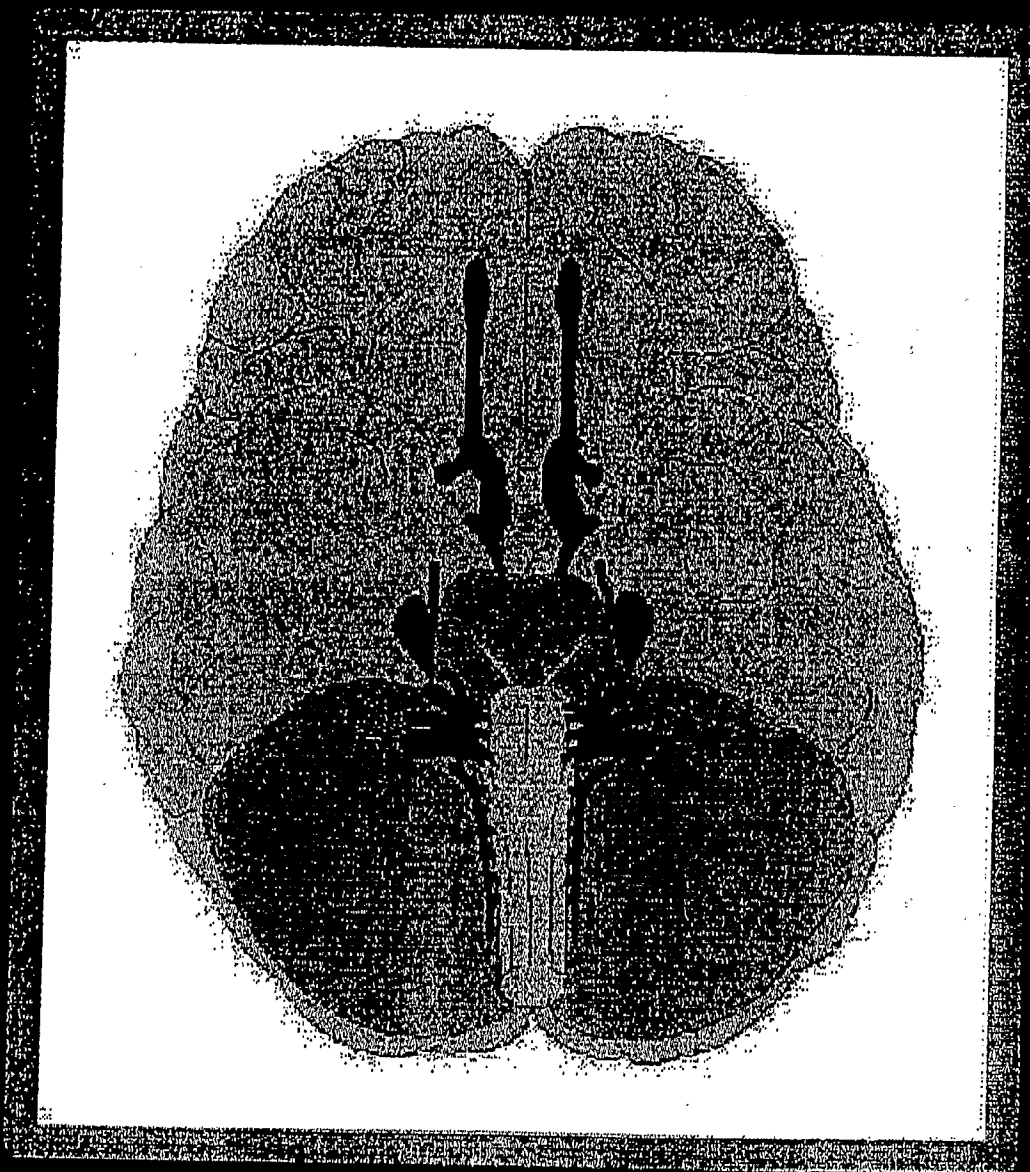


Figure 4

Soumen Mukherjee
(SOU MEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

Sample MRI of Human Brain

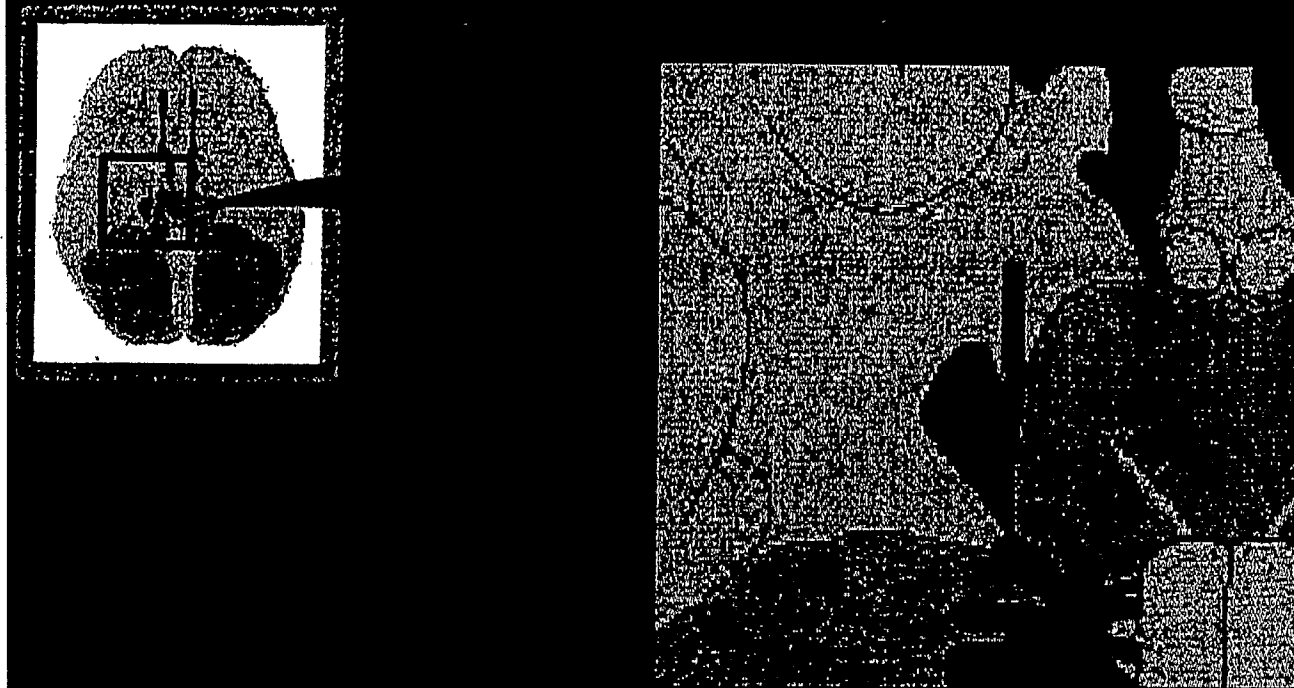


Figure 5

Soumen Mukherjee

(SOUMEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

PROVISIONAL

Sample MRI of Human Brain (Pixel View)

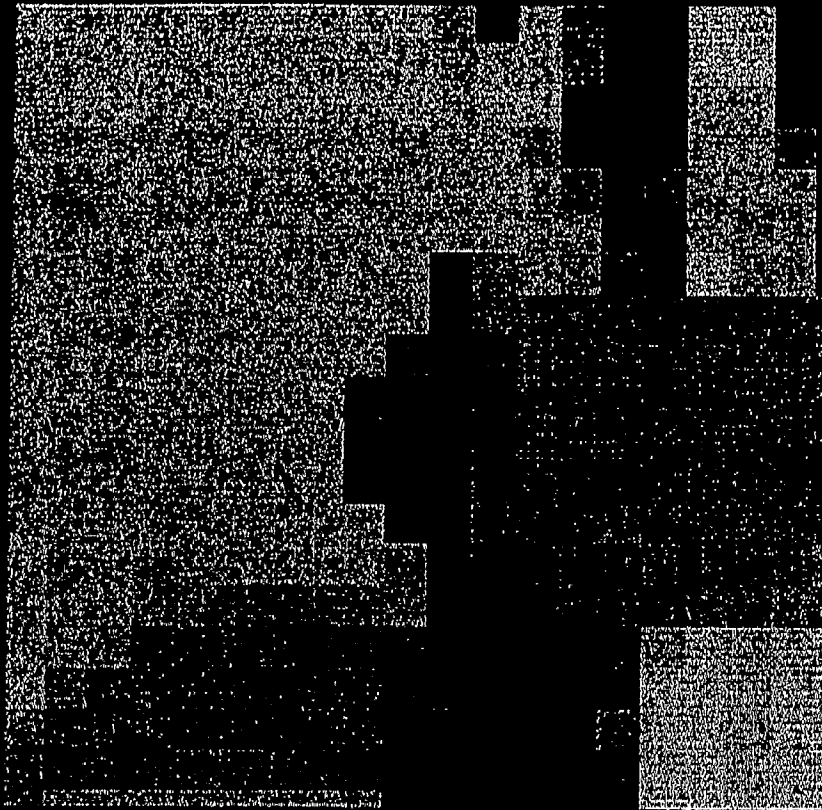


Figure 6

(Signature)
(SOUMEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

36 Pixel Region

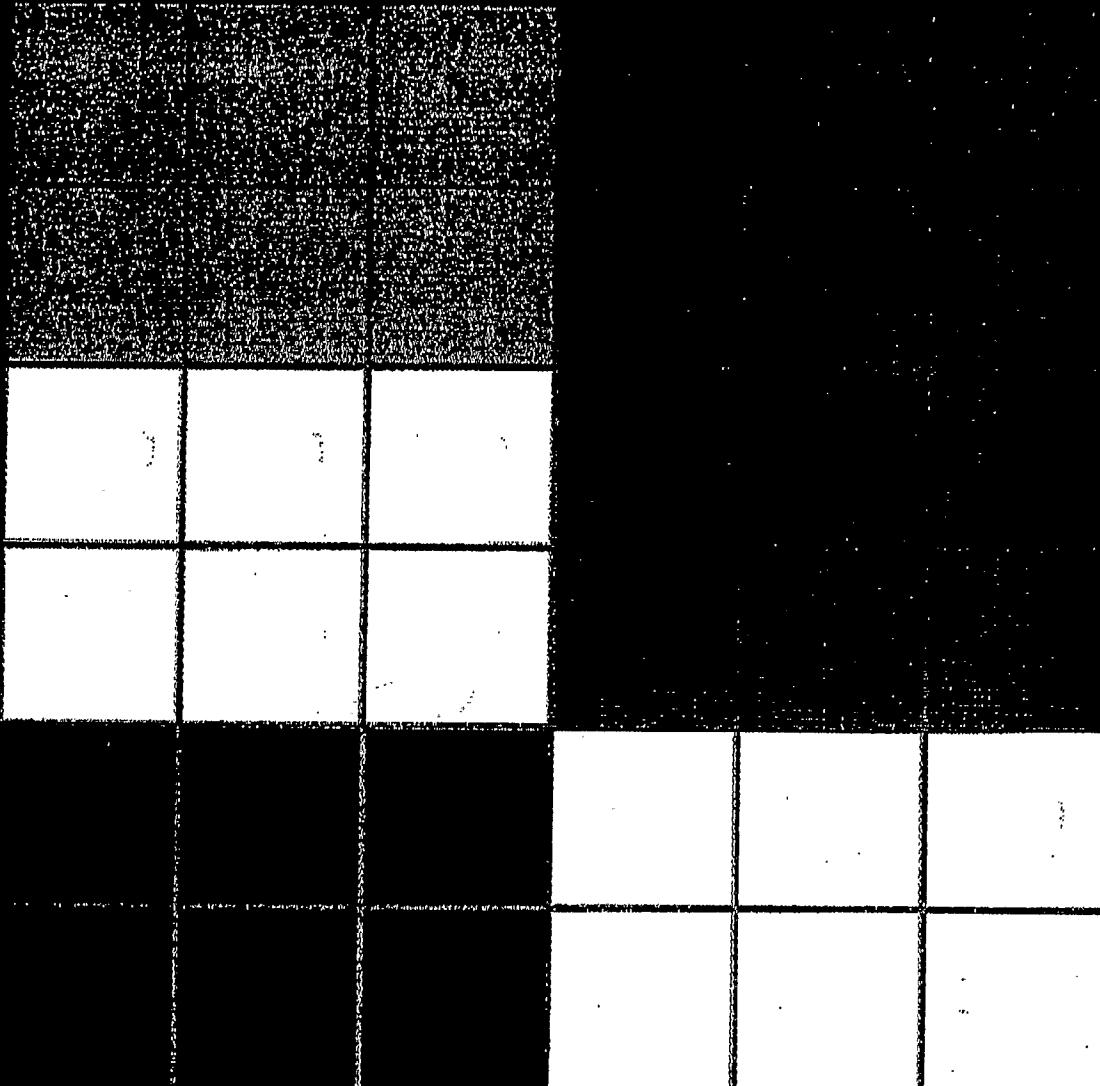


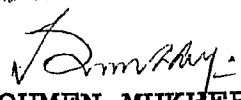
FIGURE 7

Soumen Mukherjee
(SOUMEN MUKHERJEE)
of D. P. *Agarwal & Co.*
APPLICANTS' AGENT

36 Pixel Region

150	150	150	100	100	100
150	150	150	100	100	100
150	150	150	100	100	100
250	250	250	100	100	100
250	250	250	100	100	100
20	20	20	250	250	250
20	20	20	250	250	250

FIGURE 8


(SOUMEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

PROVISIONAL



Horizontal RCC

150	150	150	150	150	150
150	150	150	150	150	150
250	250	250	250	250	250
250	250	250	250	250	250
			250	250	250
			250	250	250

Rules for adjacent pixels:

If Same value, bit plane = '1'

If Different value, bit plane = '0'

Values Stored

150	150	150	150
250	100	250	100
20	250	20	250

Bit Plane

0	1	1	0	1	1
0	1	1	0	1	1
0	1	1	0	1	1
0	1	1	0	1	1
0	1	1	0	1	1
0	1	1	0	1	1

Figure 9

(SOUMEN MUKHERJEE)

of D. P. AHUJA & CO.
APPLICANTS' AGENT

PROVISIONAL

Vertical RCC

150	150	150	250	250	250
150	150	150	250	250	250
250	250	250	250	250	250
250	250	250	250	250	250
			250	250	250
			250	250	250

Rules for adjacent pixels:
 If Same value, bit plane = '1'
 If Different value, bit plane = '0'

Values Stored

150	250	20	150
250	20	150	250
20	100	250	100
250	100	250	

Bit Plane

0	0	0	0	0	0
1	1	1	1	1	1
0	0	0	1	1	1
1	1	1	1	1	1
0	0	0	0	0	0
1	1	1	1	1	1

Figure 10

(Signature)
 (SOUMEN MUKHERJEE)
 of D. P. AHUJA & CO.
 APPLICANTS' AGENT

PROVISIONAL

Bit Plane

	1	1		1	1
0	1	1	0	1	1
	1	1	0	1	1
0	1	1	0	1	1
	1	1		1	1
0	1	1	0	1	1

Bit Plane

	0	0		0	0
1	1	1	1	1	1
	0	0	1	1	1
1	1	1	1	1	1
	0	0		0	0
1	1	1	1	1	1

Final Values Stored

150 100 250
20 250

Figure 11

(SOURNEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

Original Values

150	150	150	150	100	150	150
150	150	150	150	100	100	150
250	250	250	250	100	100	150
250	250	250	250	100	100	150
				250	250	250
				250	250	250

Total Memory Required = 288 bits


Final Values Stored

150 100 250
20 250

After RCC

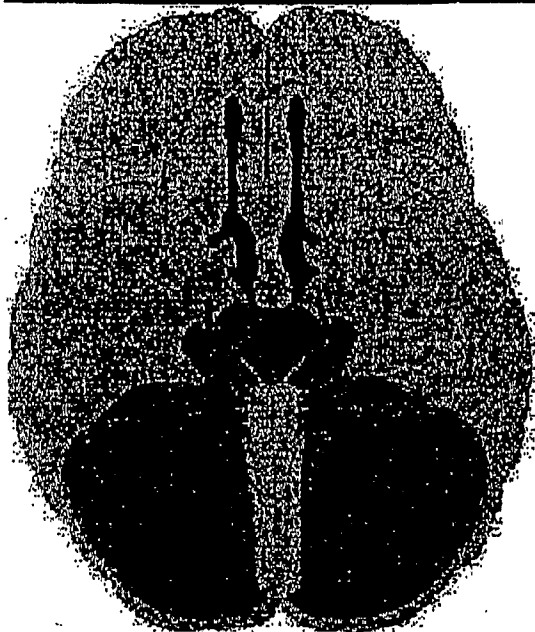
Total Memory
Required = 112 bits

Figure 12.


(SOUMEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

PROVISIONAL

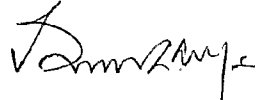
Original Image



Original File Size = 188 kb

File Size After RCC = 44 kb

Figure 13


(SOUMEN MUKHERJEE)
of D. P. AHUJA & CO.
APPLICANTS' AGENT

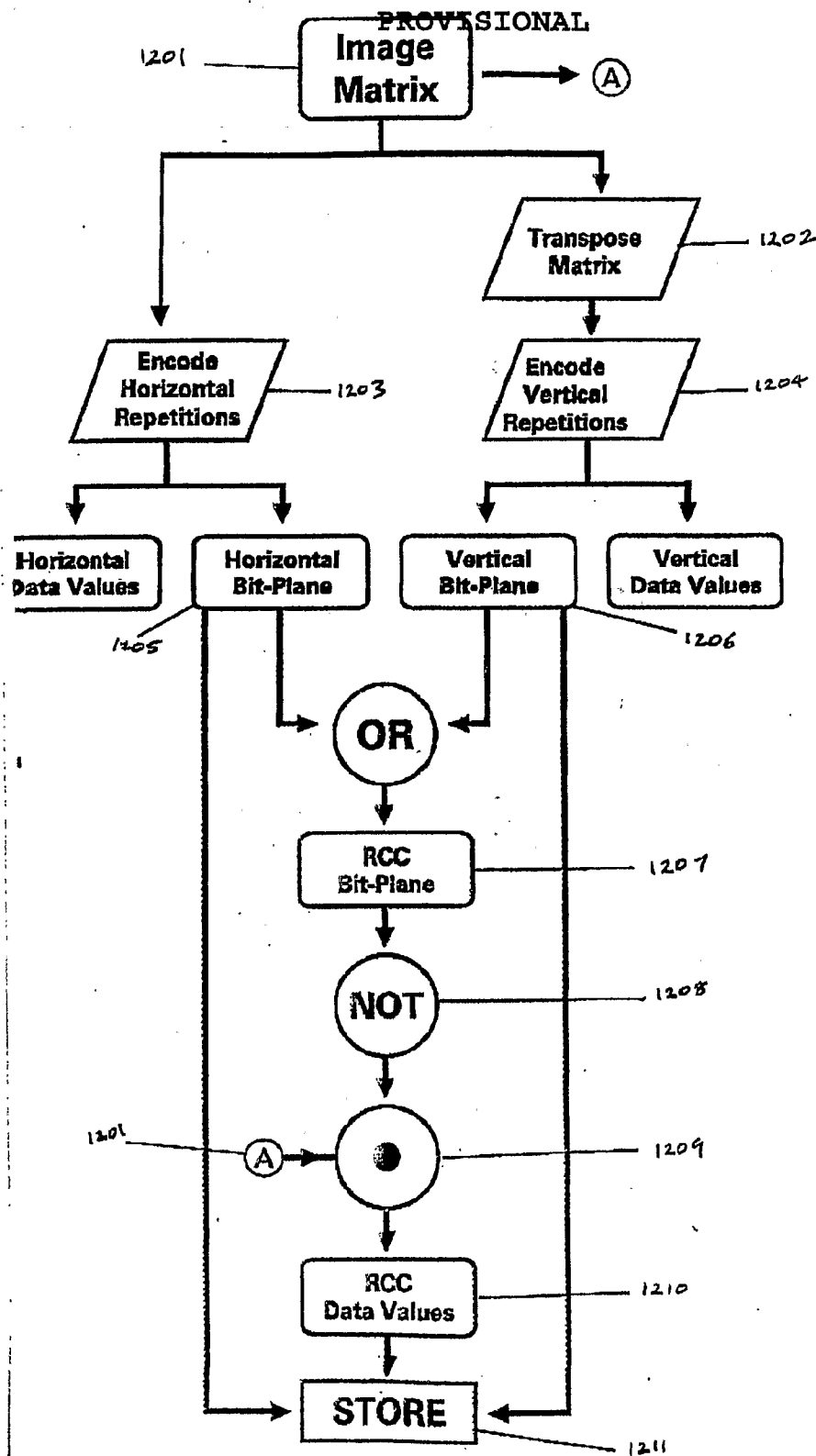


FIGURE 14

● DOT-PRODUCT

(A) CONNECTOR

Soumen Mukherjee
 (SOUMEN MUKHERJEE)
 of D. P. AHUJA & CO.
 APPLICANTS' AGENT